UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P O Box 1450
Alexandria, Virginia 22313-1450
www uspto gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/537,064	06/01/2005	Albertina De Bunje	NL 021196	8827

24737 7590 01/12/2009

PHILIPS INTELLECTUAL PROPERTY & STANDARDS
P.O. BOX 3001
BRIARCLIFF MANOR, NY 10510

EXAMINER	
SCIACCA, SCOTT M	

ART UNIT	PAPER NUMBER
2446	

MAIL DATE	DELIVERY MODE
01/12/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/537,064	DE BUNJE ET AL.	
	Examiner	Art Unit	
	Scott M. Sciacca	2446	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *03 November 2008*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-9 and 11-20* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-9 and 11-20* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *01 June 2005* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☒ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

DETAILED ACTION

This office action is responsive to communications filed on November 3, 2008. Claim 8 has been amended. Claims 1-9 and 11-20 are pending in the application.

Claim Rejections - 35 USC § 103

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

2. Claims 1, 3-5, 8, 9 and 12-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kaiserswerth et al. (US 6,195,701) in view of Silberschatz (Operating System Concepts).

Regarding Claim 1, Kaiserswerth teaches a method of scheduling schedulable component in a hard real time system for processing time dependent streams of data elements (*"The present invention concerns a method and apparatus for the synchronization and the scheduling of multiple data streams as well as for the scheduling of tasks in operating systems with hard real-time requirements"* – See Col. 1, lines 8-11), where the number of schedulable components is larger than the number of available processors for processing said components (Fig. 1 shows a single CPU **12** and Fig. 2 shows a plurality of streams – **Stream 1**, **Stream 2** and **Stream 3**) and where each of said components has at least one input and one output (*"In multimedia systems*

multiple data streams must be synchronized and scheduled for playout to, for example, a speaker 15 and a video display 14" – See Col. 2, lines 47-49) characterized in that the method comprises the steps of consecutively:

determining for each schedulable component the earliest time on which said component can contribute to the output of said hard real time system (*"In the example given in FIG. 3, a data unit from stream 1 cannot be played out before a time mark 30 (S1Start) and it must be played out before a time mark 31 (S1End)"* – See Col. 2, lines 66-67 & Col. 3, lines 1-2),

Kaiserswerth does not explicitly teach scheduling <u>only</u> the schedulable component that can contribute at the total earliest time to the output of said real time system. However, Silberschatz does teach scheduling only the schedulable component that can contribute at the total earliest time to the output of said real time system (*"6.3.2 Shortest-Job-First Scheduling"* – See p. 158; *"This algorithm associates with each process the length of the latter's CPU burst. When the CPU is available, it is assigned to the process that has the smallest next CPU burst"* – See p. 158). On p. 152, under section 6.1.1, Silberschatz shows how a plurality of schedulable components (processes) are scheduled to be handled by a processor (i.e., CPU). Each process must be processed by the CPU before the relevant information resulting from the processing of the process can be output from the system in the form of an I/O burst. Under the Shortest-Job-First scheduling algorithm shown on pages 158-159, an earliest time at which a process can contribute to the output of the system is determined by determining a "Burst Time" for all pending processes. As an example, Silberschatz

gives the burst time in milliseconds for each process P_1-P_4, as shown on pages 158-159. Each process must be handled by the CPU before it can contribute to the output of the system. According to the Shortest-Job-First algorithm, the process with the shortest burst time will be scheduled first. This way, the process which takes the least amount of processing time before it can contribute to the output of the system will be scheduled. Thus, the process which can contribute to the output of the system at the total earliest time will be scheduled first.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use the scheduling algorithm taught by Silberschatz to schedule the components taught by Kaiserswerth. Silberschatz states that "The SJF algorithm is provably optimal, in that it gives the minimum average waiting time for a given set of processes. By moving a short process before a long one, the waiting time of the short process decreases more than it increases the waiting time of the long process. Consequently, the average waiting time decreases." (See p. 159) Thus, one of ordinary skill would have been motivated to use the scheduling algorithm taught by Silberschatz to schedule the components taught by Kaiserswerth in order to guarantee a minimum average waiting time for components to be able to contribute to the output of the real time system.

Regarding Claim 3, Kaiserswerth in view of Silberschatz teaches the method of Claim 1. Kaiserswerth further teaches wherein a length of a predefined time interval is specified for each component and a component is schedulable when time stamped data

elements from said predefined time interval of said time dependent stream of time stamped data element is available at all inputs of said component (*"A stream may be synchronized to one or multiple other streams or to time stamps 23 defined by an external clock"* – See Col. 2, lines 52-54; *"A data stream is a sequence of data units. In FIG. 3, the synchronization of two data units is illustrated. The SyncPoints are translated to time marks relative to the system which processes the received data streams"* – See Col. 2, lines 61-64).

Regarding Claim 4, Kaiserswerth in view of Silberschatz teaches the method of Claim 3. Kaiserswerth further teaches wherein the availability of said predefined time interval of said time stamped data elements is determined by defining a begin time and an end time of said predefined time interval (*"In the example given in FIG. 3, a data unit from stream 1 cannot be played out before a time mark 30 (S1Start) and it must be played out before a time mark 31 (S1End)"* – See Col. 2, lines 66-67 & Col. 3, lines 1-2) and checking when the time, until which data has been processed by a preceding component, is newer than the end time of said predefined time interval (Fig. 4 illustrates checking when the time is newer than the end of the predetermined time interval, S1End).

Regarding Claim 5, Kaiserswerth in view of Silberschatz teaches the method of Claim 4. Kaiserswerth further teaches wherein the step of determining the earliest time on which said component can contribute to the output is performed by:

identifying possible paths of subsequent components that the data elements have to be processed by in order to reach the output of said system from said component (*"In multimedia systems multiple data streams must be synchronized and scheduled for playout to, for example, a speaker 15 and a video display 14, as illustrated in FIG. 1."* – See Col. 2, lines 47-49);

determining an earliest contribution time for each possible path by subtracting from the begin time of said predefined time interval the length of each of the predefined time intervals specified for each of said subsequent components in said path (*"A stream may be synchronized to one or multiple other streams or to time stamps 23 defined by an external clock"* – See Col. 2, lines 52-54; *"The SyncPoints are translated to time marks relative to the system which processes the received data streams"* – See Col. 2, lines 62-64); and

determining the earliest time on which said component can contribute to the output as the earliest determined contribution time (*"After SxStart and before SxEnd it is in the READY state"* – See Col. 3, lines 6-7).

Regarding Claim 8, Kaiserswerth teaches a system, comprising:

a processor device configured to:

determine an execution time, for each of a plurality of schedulable components, at which an output of each schedulable component is able to be processed by the system (*"In the example given in FIG. 3, a data unit from stream 1 cannot be played out before a time mark 30 (S1Start) and it must be played out before a time mark 31*

(S1End)" – See Col. 2, lines 66-67 & Col. 3, lines 1-2; *"SyncPoints are translated to time marks relative to the system which processes the received data streams. The SyncPoints might be translated to time marks by means of a clock or counter of said system"* – See Col. 2, lines 62-66), wherein a component is schedulable only if the component has processed all data elements with time stamps in a corresponding processing time interval (*"Before SxStart of a data stream x has been reached, the respective data unit is in the WAIT state. After SxStart and before SxEnd it is in the READY state"* – See Col. 3, lines 5-7).

Kaiserswerth does not explicitly teach scheduling processing of the output of *only one* of the schedulable components by the system based on the execution times of the plurality of schedulable components, wherein only a schedulable component that can contribute at a total earliest time to the output of said system is scheduled.

However, Silberschatz does teach scheduling only the schedulable component that can contribute at the total earliest time to the output of said real time system (*"6.3.2 Shortest-Job-First Scheduling"* – See p. 158; *"This algorithm associates with each process the length of the latter's CPU burst. When the CPU is available, it is assigned to the process that has the smallest next CPU burst"* – See p. 158). On p. 152, under section 6.1.1, Silberschatz shows how a plurality of schedulable components (processes) are scheduled to be handled by a processor (i.e., CPU). Each process must be processed by the CPU before the relevant information resulting from the processing of the process can be output from the system in the form of an I/O burst. Under the Shortest-Job-First scheduling algorithm shown on pages 158-159, an earliest

time at which a process can contribute to the output of the system is determined by determining a "Burst Time" for all pending processes. As an example, Silberschatz gives the burst time in milliseconds for each process P_1-P_4, as shown on pages 158-159. Each process must be handled by the CPU before it can contribute to the output of the system. According to the Shortest-Job-First algorithm, the process with the shortest burst time will be scheduled first. This way, the process which takes the least amount of processing time before it can contribute to the output of the system will be scheduled. Thus, the process which can contribute to the output of the system at the total earliest time will be scheduled first.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use the scheduling algorithm taught by Silberschatz to schedule the components taught by Kaiserswerth for the same reasons as those given with respect to Claim 1.

Regarding Claim 9, Kaiserswerth in view of Silberschatz teaches the system of Claim 8. Kaiserswerth further teaches the component not being schedulable if the component has not processed all of the data elements with time stamps in the corresponding processing time interval (*"Before SxStart of a data stream x has been reached, the respective data unit is in the WAIT state"* – See Col. 3, lines 5-6).

Regarding Claim 12, Kaiserswerth in view of Silberschatz teaches the system of Claim 8. Kaiserswerth further teaches the data elements being from a data stream

(*"The present invention concerns a method and an apparatus for the synchronization and the scheduling of multiple data streams and real time tasks"* – See Abstract).

Regarding Claim 13, Kaiserswerth in view of Silberschatz teaches the system of Claim 8. Kaiserswerth further teaches the processing time interval being a predefined time box with a start time and an end time (*"a data unit from stream 1 cannot be played out before a time mark 30 (S1Start) and it must be played out before a time mark 31 (S1End)"* – See Col. 2, lines 66-67 & Col. 3, lines 1-2).

Regarding Claim 14, Kaiserswerth in view of Silberschatz teaches the system of Claim 8. Kaiserswerth further teaches the data elements being produced by a preceding component (*"According to this example, two data streams are received by means 18 for extraction of time marks. The first data stream originates from a storage disk 16, whereas the second data stream is sent via a network 17 to said means 18"* – See Col. 3, lines 38-42).

Regarding Claim 15, Kaiserswerth in view of Silberschatz teaches the system of Claim 8. Kaiserswerth further teaches the schedulable component being a self-contained part of the system, performing a sub-task that is atomic (Storage disk 16 is part of the system disclosed by Kaiserswerth. It performs the sub-task of storing data).

Regarding Claim 16, Kaiserswerth in view of Silberschatz teaches the system of Claim 8. Kaiserswerth further teaches the system being a hard real time system for processing time dependent streams of data elements (*"The present invention concerns a method and an apparatus for the synchronization and the scheduling of multiple data streams and real time tasks"* – See Abstract).

Regarding Claim 17, Kaiserswerth in view of Silberschatz teaches the system of Claim 8. Kaiserswerth teaches the execution time being based on algorithmic time and is converted to real time once the output is processed (*"The SyncPoints might be translated to time marks by means of a clock or counter of said system"* – See Col. 2, lines 64-66).

Regarding Claim 18, Kaiserswerth teaches a method, comprising:

defining a current time box for each of a plurality of components, wherein each current time box has a start time and an end time, and each component processes data elements in at least one corresponding current time box (*"In the example given in FIG. 3, a data unit from stream 1 cannot be played out before a time mark 30 (S1Start) and it must be played out before a time mark 31 (S1End)"* – See Col. 2, lines 65-66 & Col. 3, lines 1-2); and

scheduling a first of the plurality of the components for execution when all data elements with time stamps in the first component's current time box are present, wherein all of the data elements for the first component are present in the first

component's current time box before all data elements for another one of the plurality of components are present in a corresponding current time box (*"IF both data units are in the READY state THEN they can be played out"* – See Col. 3, lines 14-15; Fig. 2 shows a plurality of streams originating from separate components arriving at different points in time. Thus, all data elements from a first component are present before all data elements from a second component).

Kaiserswerth does not explicitly teach scheduling *only* the schedulable component that can contribute at a total earliest time to the output.

However, Silberschatz does teach scheduling only the schedulable component that can contribute at the total earliest time to the output of said real time system (*"6.3.2 Shortest-Job-First Scheduling"* – See p. 158; *"This algorithm associates with each process the length of the latter's CPU burst. When the CPU is available, it is assigned to the process that has the smallest next CPU burst"* – See p. 158). On p. 152, under section 6.1.1, Silberschatz shows how a plurality of schedulable components (processes) are scheduled to be handled by a processor (i.e., CPU). Each process must be processed by the CPU before the relevant information resulting from the processing of the process can be output from the system in the form of an I/O burst. Under the Shortest-Job-First scheduling algorithm shown on pages 158-159, an earliest time at which a process can contribute to the output of the system is determined by determining a "Burst Time" for all pending processes. As an example, Silberschatz gives the burst time in milliseconds for each process P_1-P_4, as shown on pages 158-159. Each process must be handled by the CPU before it can contribute to the output

of the system. According to the Shortest-Job-First algorithm, the process with the shortest burst time will be scheduled first. This way, the process which takes the least amount of processing time before it can contribute to the output of the system will be scheduled. Thus, the process which can contribute to the output of the system at the total earliest time will be scheduled first.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use the scheduling algorithm taught by Silberschatz to schedule the components taught by Kaiserswerth for the same reasons as those given with respect to Claim 1.

Regarding Claim 19, Kaiserswerth in view of Silberschatz teaches the method of Claim 18. Kaiserswerth further teaches the data elements being from a stream of data elements in which each data element in the stream is time stamped (Fig. 2 shows a plurality of streams with timestamps 23).

Regarding Claim 20, Kaiserswerth in view of Silberschatz teaches the method of Claim 1. Kaiserswerth further teaches each of the schedulable components having a corresponding earliest time at which it can contribute to the output of the real time system, and wherein the total earliest time is an earliest of the earliest times of the schedulable components (*"a data unit from stream 1 cannot be played out before a time mark 30 (S1Start)"* – See Col. 2, lines 65-66 & Col. 3, line 1; The total earliest time

belongs to a stream having an S1Start value that is earlier in time than the rest of the streams).

3. Claims 2 and 11 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kaiserswerth et al. (US 6,195,701) in view of Silberschatz (Operating System Concepts) and further in view of Kamiya (US 2001/0026558).

Regarding Claim 2, Kaiserswerth and Silberschatz do not explicitly teach the method wherein if a number of schedulable components contribute to the output of said real time system at the same total earliest time, then scheduling of said number of components is performed using push scheduling. However, Kamiya does teach scheduling components using push scheduling (*"there is provided a distributed pipeline scheduling method for a system"* – See [0031]). It would have been obvious to one of ordinary skill in the art at the time the invention was made to use push scheduling for components that contribute to the output of a real time system at the same earliest time. One would have been motivated to do so in order to provide a mechanism for dealing with components that are simultaneously in contention for the output of the real time system.

Regarding Claim 11, Kaiserswerth and Silberschatz do not explicitly teach push scheduling being employed when two of the plurality of schedulable components have the earliest execution time.

However, Kamiya does teach push scheduling being employed when two of the plurality of schedulable components have the earliest execution time (*"there is provided a distributed pipeline scheduling method for a system"* – See [0031]).

It would have been obvious to use push scheduling for the same reasons as those given with respect to Claim 2.

4. Claims 6 is rejected under 35 U.S.C. 103(a) as being unpatentable over Kaiserswerth et al. (US 6,195,701) in view of Silberschatz (Operating System Concepts) and further in view of Vogl et al. (7,150,017).

Regarding Claim 6, Kaiserswerth in view of Silberschatz teaches the method of Claim 4. Kaiserswerth further teaches wherein the step of determining the earliest time on which said component can contribute to the output is performed by:

identifying possible paths of subsequent components that the data elements have to be processed by in order to reach the output of said system from said component (*"In multimedia systems multiple data streams must be synchronized and scheduled for playout to, for example, a speaker 15 and a video display 14, as illustrated in FIG. 1."* – See Col. 2, lines 47-49);

determining an earliest contribution time for each possible path by subtracting from the begin time of said predefined time interval, the length of each of the predefined time intervals specified for each of said subsequent components in said path *("A stream may be synchronized to one or multiple other streams or to time stamps 23 defined by*

an external clock" – See Col. 2, lines 52-54; *"The SyncPoints are translated to time marks relative to the system which processes the received data streams"* – See Col. 2, lines 62-64); and

determining the earliest time on which said component can contribute to the output as the earliest determined contribution time (*"After SxStart and before SxEnd it is in the READY state"* – See Col. 3, lines 6-7).

Kaiserswerth and Silberschatz do not explicitly teach predefined time intervals having been subtracted a displacement value. Vogl teaches time intervals having a displacement value (*"In an alternate embodiment, the duration 220 field could hold a number which indicated an offset (perhaps in seconds) against the release time 210"* – See Col. 7, lines 49-52).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to subtract a displacement value from predefined time intervals in order to adjust the time interval wherein a component will contribute to the output of a real time system.

5. Claims 7 is rejected under 35 U.S.C. 103(a) as being unpatentable over Kaiserswerth et al. (US 6,195,701) in view of Silberschatz (<u>Operating System Concepts</u>) and further in view of Willard (US 6,374,405).

Regarding Claim 7, Kaiserswerth in view of Silberschatz teaches the method of Claim 1. Kaiserswerth and Silberschatz do not explicitly teach none of the other

schedulable components being scheduled until after the scheduled schedulable component is processed and contributes to the output of the real time system. However, Willard does teach none of the other schedulable components being scheduled until after the scheduled schedulable component is processed and contributes to the output of the real time system (*"In FIG. 7a, all of the packets of a single module (Mod. 1) are to transmitted consecutively"* – See Col. 9, lines 21-22; *"A second module (Mod. 2) can be scheduled in the same manner for transmission after the first module"* – See Col. 9, lines 33-35; The scheduling Willard's disclosure relates to scheduling modules of data for output from a source to a receiver).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to schedule a next component only after the current component is finished being processed and has contributed to the output of a real time system. Motivation for doing so would be to ensure that modules are delivered by a particular time.

Response to Arguments

6. Applicant's arguments filed on November 3, 2008 have been fully considered but they are not persuasive.

On page 11 of the remarks, Applicant argues "It is respectfully submitted that Silberschatz (emphasis added) 'associates with each process the length of the latter's next CPU burst. <u>When</u> the <u>CPU</u> is <u>available, it is assigned to the process</u>

<u>that has the smallest next CPU burst</u>.' (See, Silberschatz, page 158, section 6.3.2, lines 2-4.) Silberschatz makes clear that a (emphasis added) 'more appropriate term [for the scheduling of Silberschatz] would be the shortest next CPU burst, because the <u>scheduling is done by examining the length of the next CPU burst of a process</u> ...'"

Silberschatz does state that the scheduling of processes is based on the burst length of a process. However, Examiner will show below how the scheduling algorithm for scheduling the execution of multiple processes on a single CPU is equivalent to "scheduling only the schedulable component that can contribute at the total earliest time to the output of said real time system" in the presently claimed invention.

On pages 11-12 of the remarks, Applicant argues "Accordingly, Silberschatz merely schedules components based on the CPU burst time of the components <u>irrespective of whether the components contribute to the output of the real time system</u>. The Office Action seems to recognize this distinction in that the Office Action cites Silberschatz in stating that (emphasis added) '[w]hen the CPU is available, <u>it is assigned to the process that has the smallest next CPU burst</u>' (See, Office Action, page 4, lines 14-15.) However, then the Office Action enters into unsupported conjecture in finding that (emphasis added) '<u>under, the Shortest-Job-First scheduling</u> algorithm shown on pages 158-159, <u>an earliest time at which</u>

a process can contribute to the output of the system is determined by determining a 'Burst Time' for all pending processes.'"

Examiner's statement that "under, the Shortest-Job-First scheduling algorithm shown on pages 158-159, an earliest time at which a process can contribute to the output of the system is determined by determining a 'Burst Time' for all pending processes" is not unsupported conjecture. This statement is supported by section **6.1.1 CPU-I/O Burst Cycle** found on pages 152-153 of Silberschatz.

Summarizing the previously cited material from Silberschatz, for example with respect to Claim 1, Silberschatz teaches a plurality of processes which are executed by sharing processing time on a single CPU.

On page 152, Silberschatz states *"The success of CPU scheduling depends on the following observed property of processes: Process execution consists of a cycle of CPU execution time and I/O wait. Processes alternate between these two states. Process execution begins with a CPU burst. That is followed by an I/O burst, then another CPU burst, then another I/O burst, and so on."* In other words, before a process can contribute to the output of the system by performing an I/O burst, a CPU execution burst must first be performed on the process.

The scheduling algorithm of Silberschatz works by scheduling a batch of processes for execution by a single CPU, giving priority to processes which require the least amount of processing time. Based on the fact that a process cannot perform an

I/O burst until it has been executed by the CPU, a process having the shortest CPU burst time will be able to contribute to the output of the system at the earliest time.

In the example of the Shortest-Job-First scheduling algorithm taught by Silberschatz, only a single CPU is utilized for executing the plurality of processes (P_1-P_4 in the example shown in section **6.3.2** on pages 158-159). Since only one process can be scheduled at a time and the process having the shortest burst time (and consequently, the one that is able to contribute to the output of the system at the earliest time) is scheduled first, then only the process that can contribute at the total earliest time to the output of the system is scheduled.

This reasoning applies to Claims 1, 8 and 18.

Conclusion

7. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Scott M. Sciacca whose telephone number is (571) 270-1919. The examiner can normally be reached on Monday thru Friday, 7:30 A.M. - 5:00 P.M. EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Jeff Pwu can be reached on (571) 272-6798. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Scott M. Sciacca/
Examiner, Art Unit 2446

/Jeffrey Pwu/
Supervisory Patent Examiner, Art Unit 2446